



Нецветаев Тимофей, Head of DevOps
Власов Юрий, архитектор DevOps



Кто мы?



Власов Юрий
Архитектор DevOps, CDEK

Нецветаев Тимофей
Head of DevOps, CDEK



О компании CDEK

- **22 года** работы
- **27** стран
- **400 000** отправок в сутки
- **50 000** сотрудников
- **1 500 000+** клиентов
- **в ТОП-10** франшиз 2021*

(по версии Businessmens.ru)



О чем этот доклад?

Выбор кролика

Выращивание кролика

Сгоревший кролик

Обновленный кролик

“ Источник нашей мудрости – наш опыт.
Источник нашего опыта – наша глупость

Александр Жорж Пьер Гитри

О чем мы НЕ будем говорить?

~~Основные понятия~~

~~Сравнение брокер-серверов~~



Кролика причесать!



1С - БИТ
Реестр НП
Сотрудник
ОХД
Pentaho
Структура компании
Пакет клиентских документов
Балансер BI
Superset
ОМП
ВПП
Договор
CRM Creatio
Контрагент
Банк
Балансер коммерции
ЛК ЮЛ
Схемы оповещений
Претензии
Платформа интеграции
RabbitMQ
Задолженность по заказам
PoS
ЦОД
Сеть
Балансер ЭК5
Управление уведомлениями
Авторизация ЭК5 (PDP)
Фронт ЭК5
Модуль сбора паспортных данных
СДЭК финанс
ДЗНП
FLO
Сервис согласования доставки
Печать заказов
Геосервис
Тарифный калькулятор
Зарплата курьера
Биллинг
Преалерт
Сортер
Отправки
Склад
Адресное хранение
Поддержка курьера
3PL
Сервис статусов
Cdek.market
CDEK ID
Транспорт
Mail Forward
Мобильное приложение
Наложка C2C
Международные сайты
OMNI
Программа лояльности
Отслеживание заказа
Балансер маркетинга
ЛК ФЛ
Keycloak
Сайт

- **Около 350** сервисов
- **Более 30** крупных web-проектов
- TPS на базах **больше 120K**
- **900 релизов** в месяц, в **любое время**

Почему не Kafka?

"**Кролики — это не только ценный мех, но и три — четыре килограмма диетического, легкоусвояемого мяса**

Франц Кафка

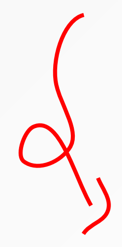


Почему не Kafka?



ERP Events

Kafka



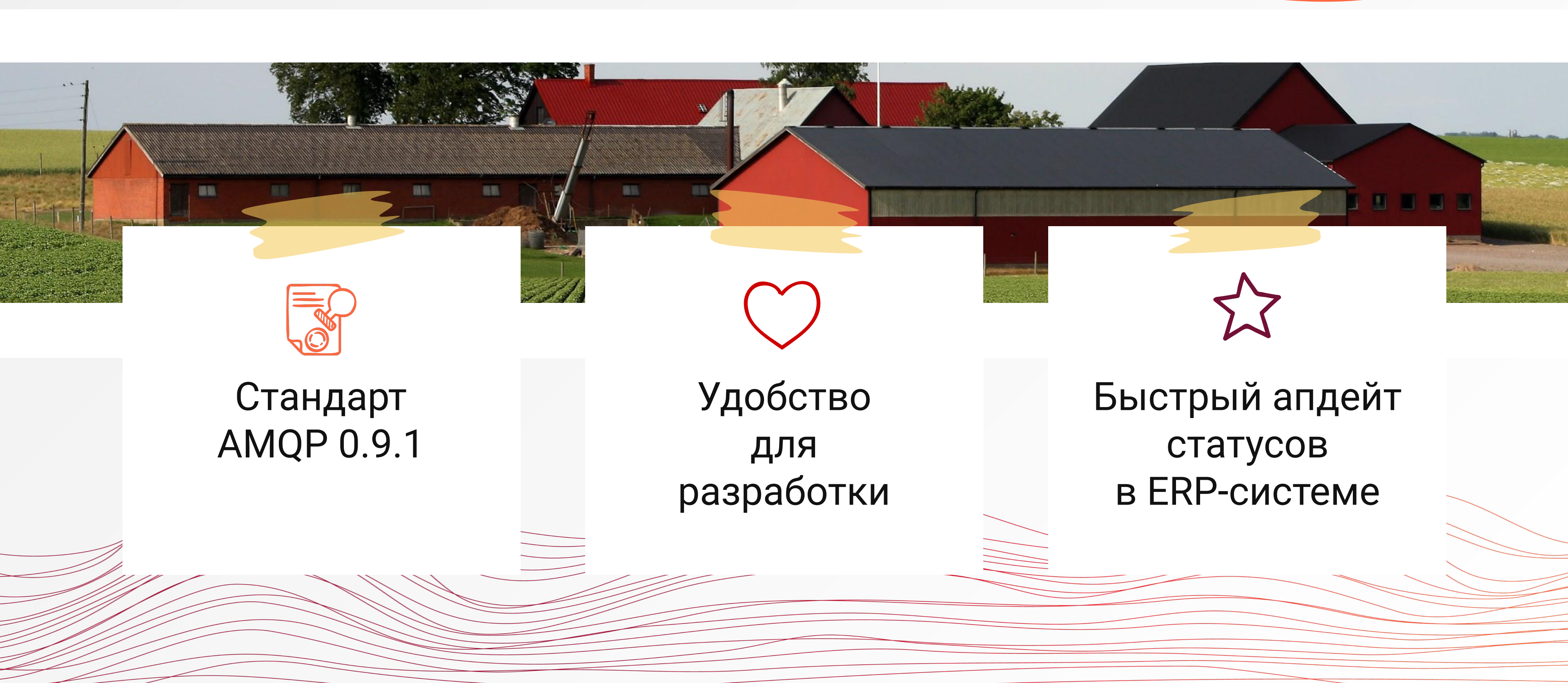
RDW, Data-lake, logs

Почему мы используем RabbitMQ?

Такой здоровый с..а !



Почему мы используем RabbitMQ?



Стандарт
AMQP 0.9.1

Удобство
для
разработки

Быстрый апдейт
статусов
в ERP-системе

Пример обновления статуса заказа

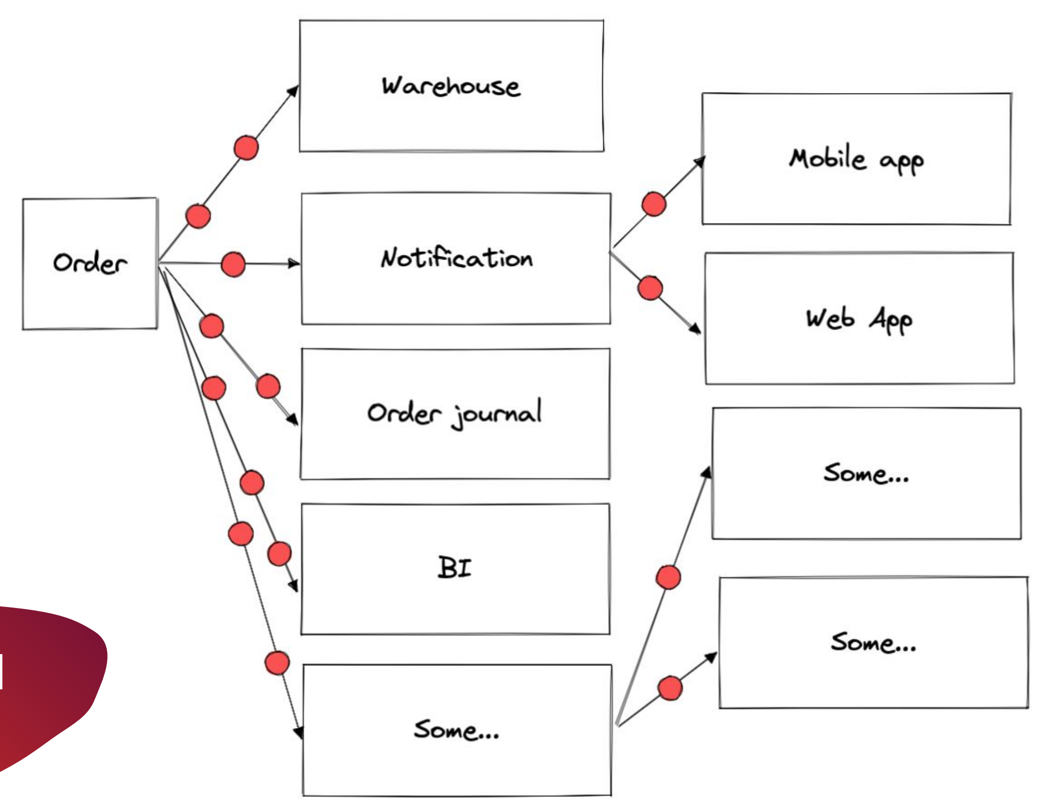


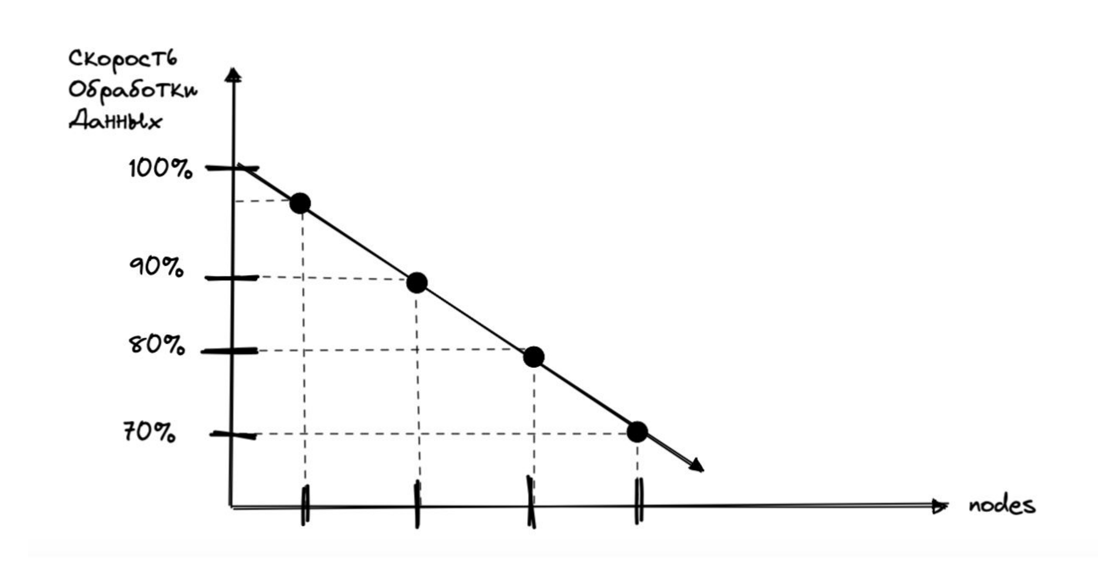
Схема до безобразия упрощена...

Почему мы отказались от HA-cluster



Плодимся, парни!

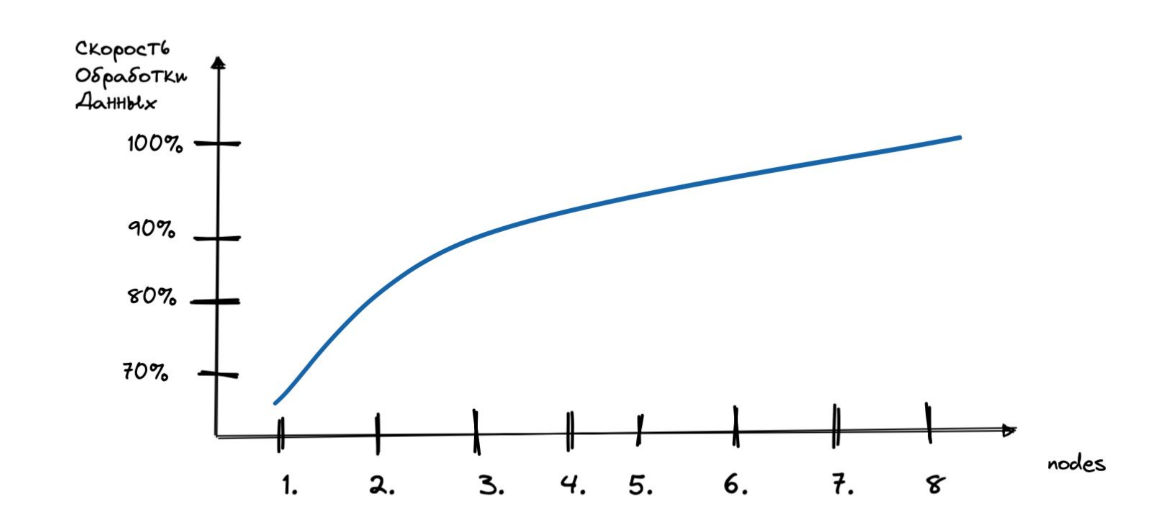
Развитие нашего кластера — HA

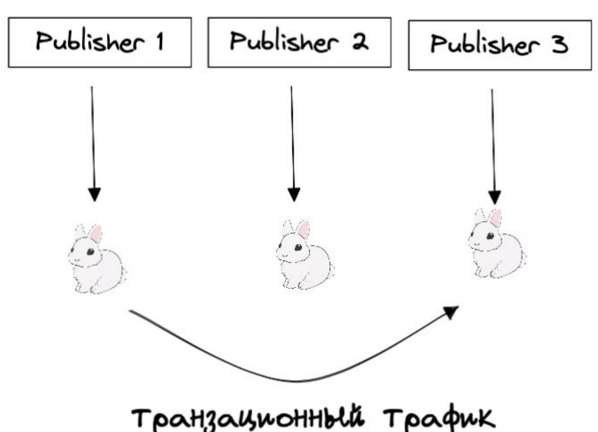




Развитие нашего кластера – Master cluster





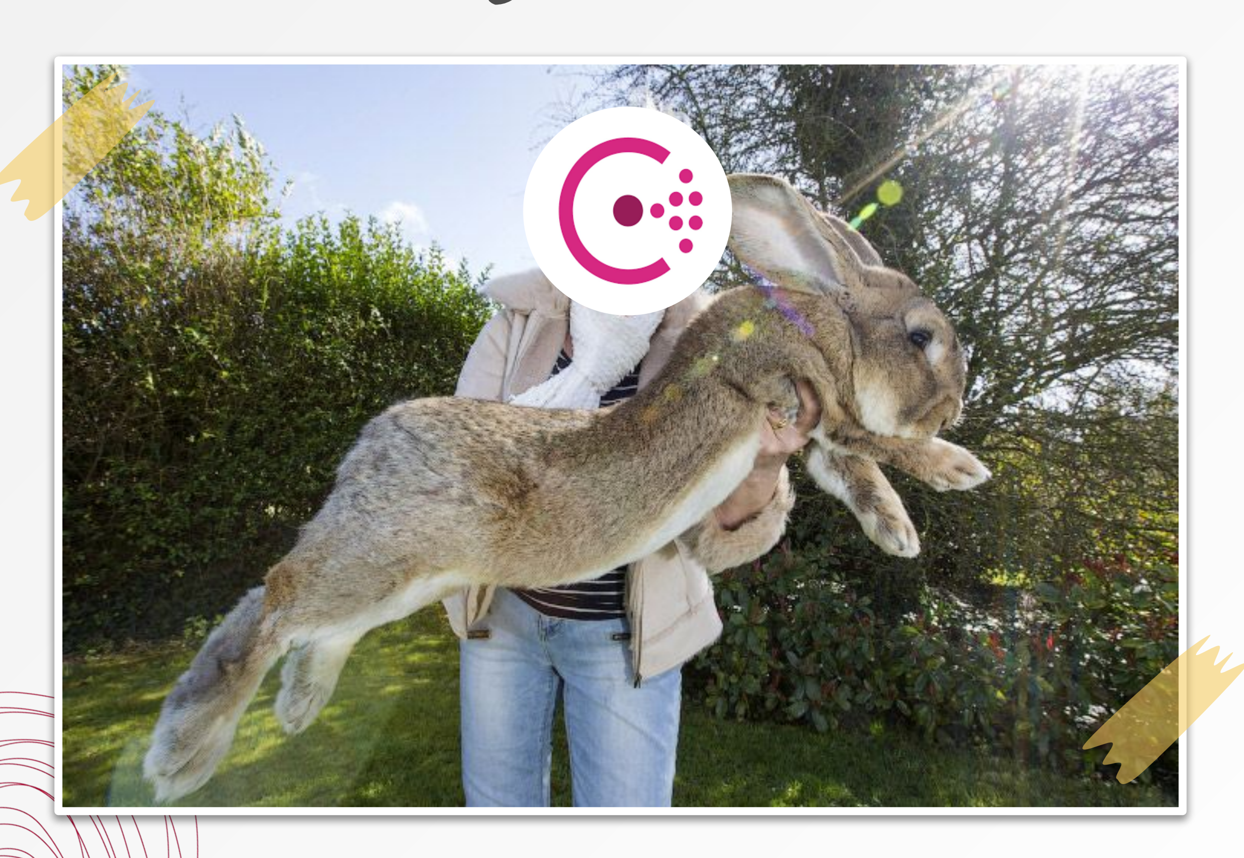


Особенности нашей кластеризации



Oh shi...

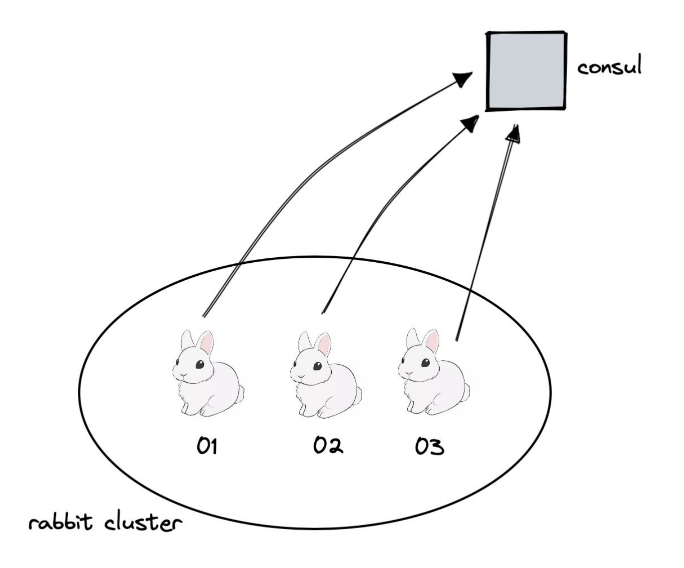
Consul Discovery

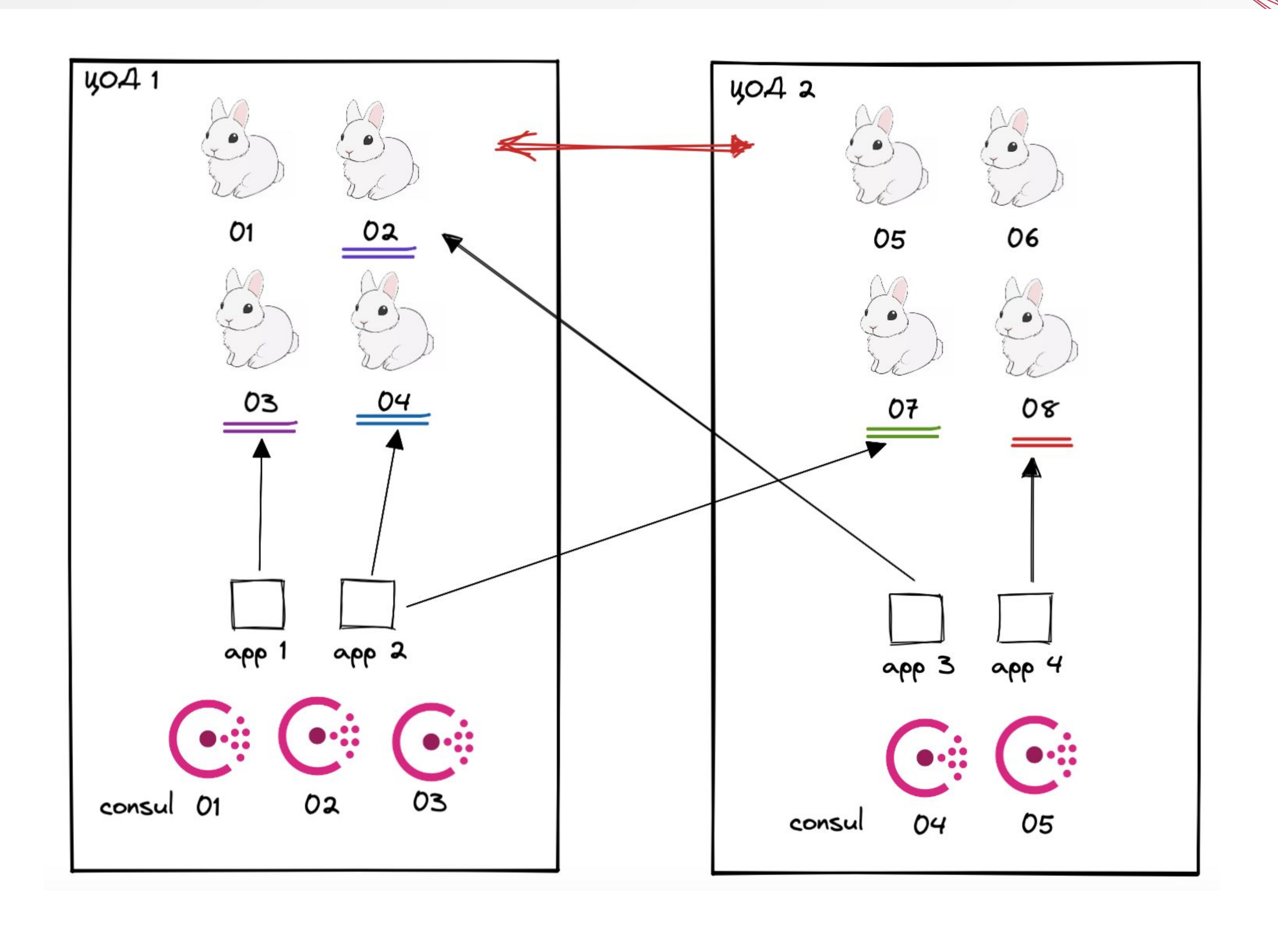


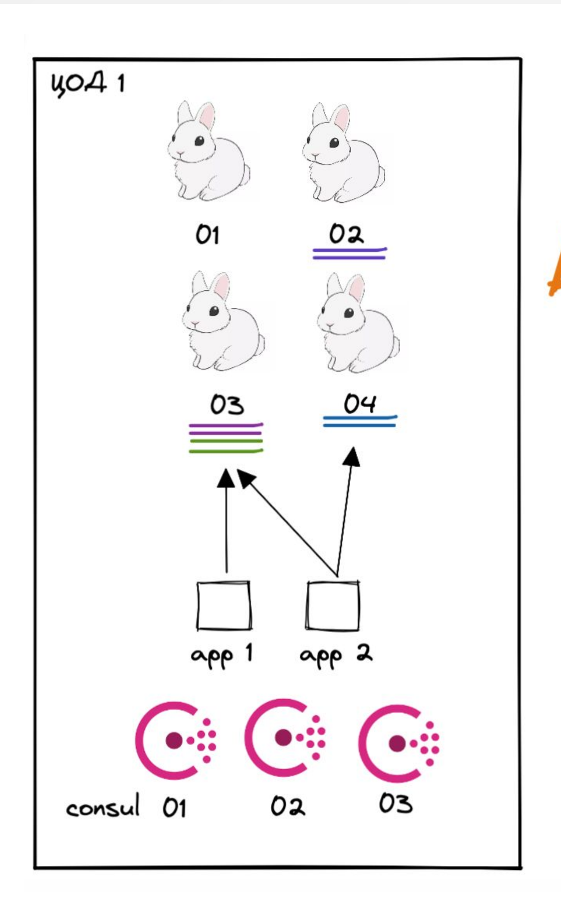
Особенности геокластеризации — Consul

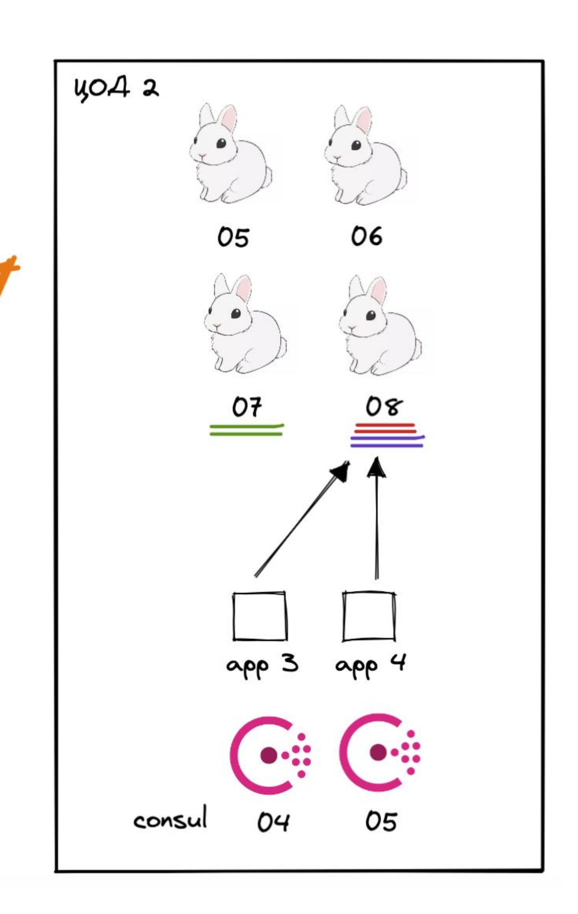


```
cluster_formation.peer_discovery_backend = consul
cluster_formation.consul.host = localhost
cluster_formation.consul.port = 8500
cluster_formation.consul.scheme = http
cluster_formation.consul.acl_token = ib_vse_videt
cluster_formation.consul.svc = rabbitmq
cluster_formation.consul.use_longname = true
cluster_formation.consul.svc_port = 5672
cluster_formation.consul.svc_tags.1 = production
cluster_formation.consul.svc_ttl = 40
cluster_formation.consul.deregister_after = 90
cluster_formation.consul.domain_suffix = consul.tech
```

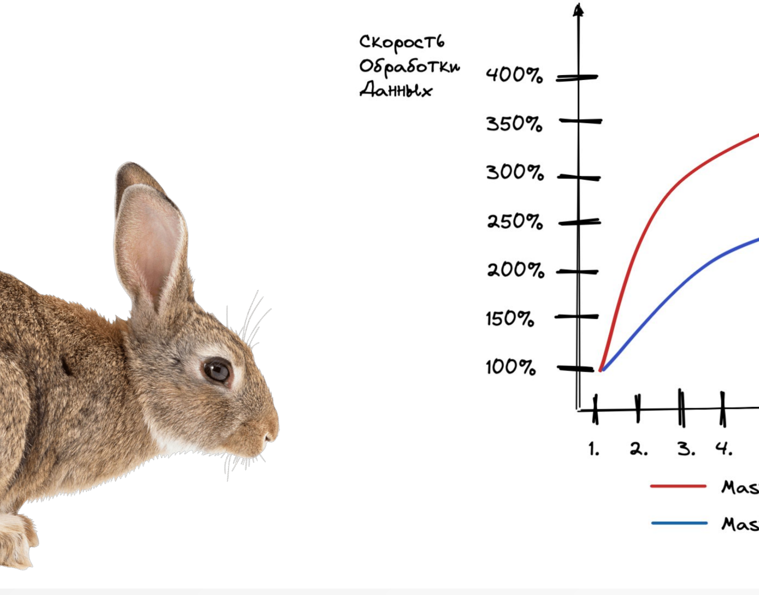


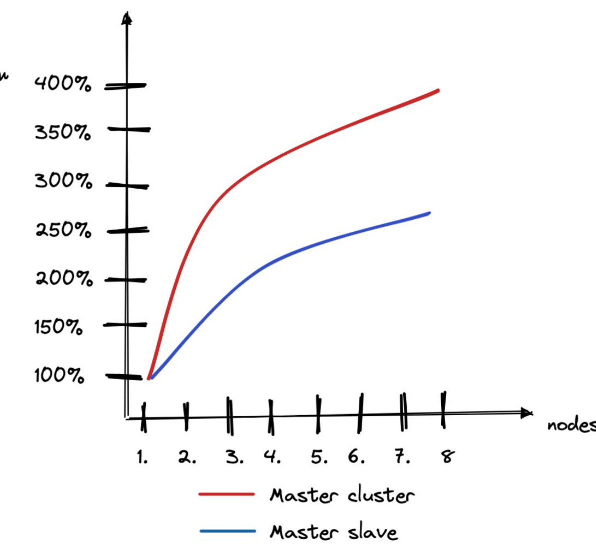

ЦОД 1
01
02
03
04
app 1
app 2
consul 01
02
03
ЦОД 2
05
06
07
08
app 3
app 4
consul
04
05

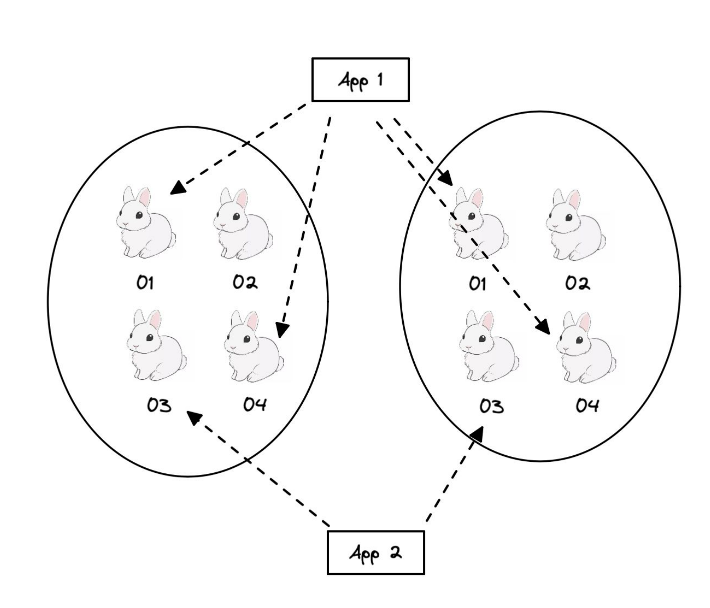

ЦОД 1
01
02
03
04
app 1
app 2
consul
01
02
03


ЦОД 2
05
06
07
08
app 3
app 4
consul
04
05

Будущее — быстрый кластер + зеркалирование







Как мы управляем очередями



Перекосы на нодах



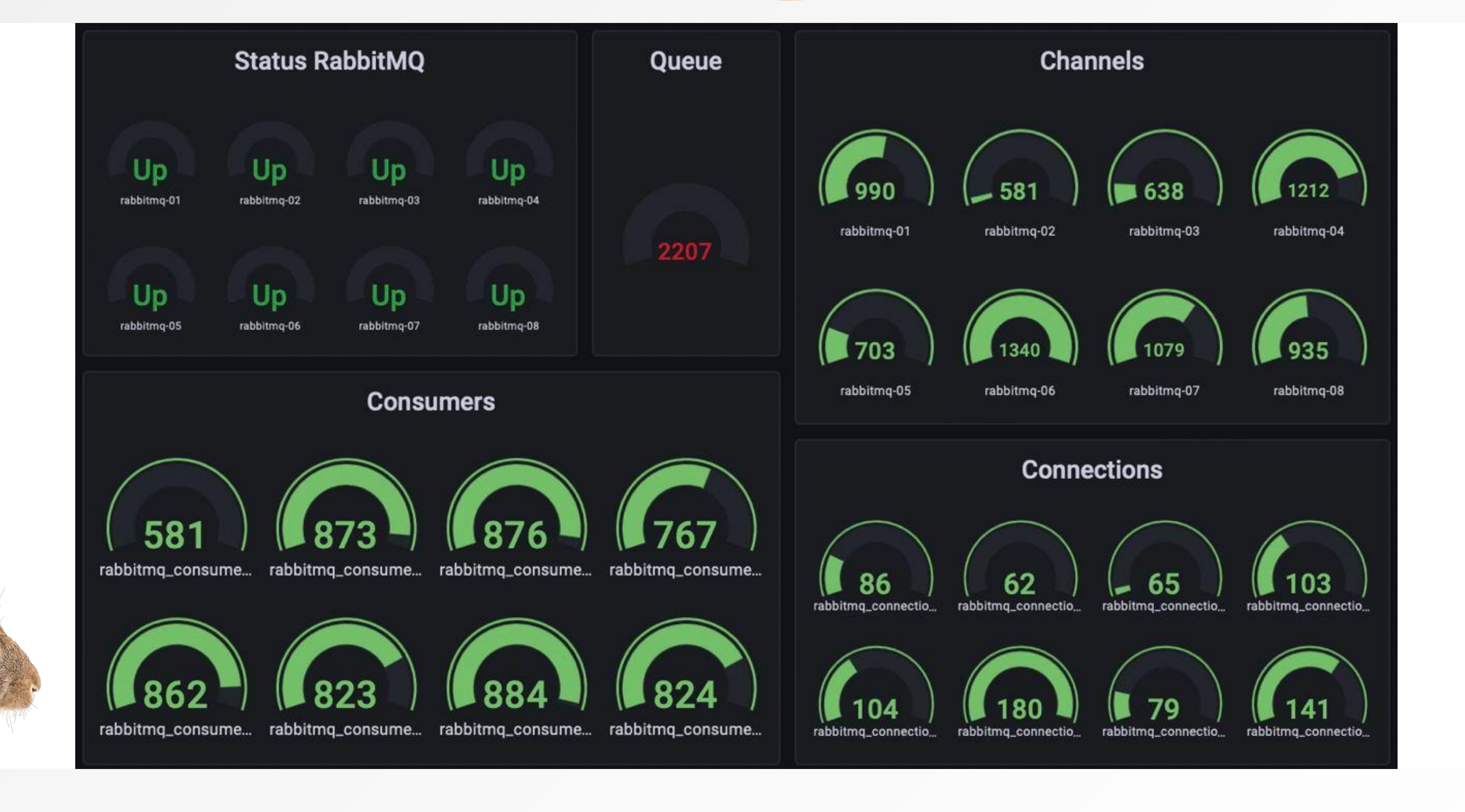
Наша система именований

Именование топика

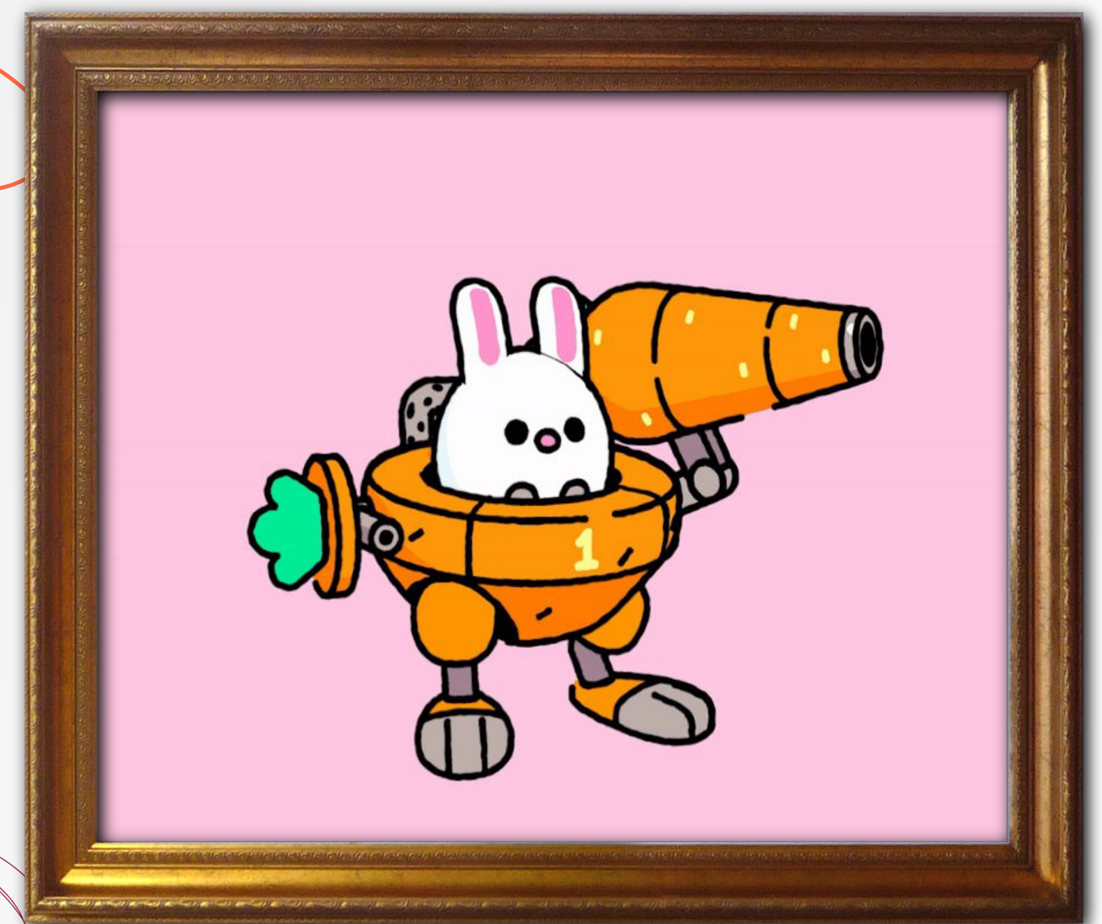
Формат именования топика предлагается следующий:
<DOMAIN>.<SUBDOMAIN>[.<SUBDOMAIN>].<OBJECT | EVENT>[_<VERSION>].<INT | EXT>[.<STAGE>]

[] - обозначены необязательные части.
DOMAIN - бизнес-домен - именно бизнес-направление деятельности компании.
SUBDOMAIN - не более 3-х вложений. Более узкоспециализированная подобласть бизнеса.
OBJECT - наименование объекта в случае публикации событий объекта.
EVENT - название события в прошедшем времени в случае публикации отдельных событий.
VERSION - номер версии события, если меняется транспорт. Является частью названия события.
INT | EXT - внутренее или внешнее событие.
STAGE - название среды. Нужно только для сред разработки, где есть общий брокер сообщений на несколько сред.

Применение политик на нодах



Тюнинг кролика



Это
же
заяц :(

Тюнинг кролика — настройка

rabbitmq-env.conf

```
RABBITMQ_SERVER_ADDITIONAL_ERL_ARGS="
+S 4:4 +sbwt none +sbwtdio none"
```

+S 4:4 — позволяет указать принудительное количество шедулеров в системе
+sbwt none +sbwtdio none — отключает перевод памяти в состояния “занят” и “ожидание”

* Актуально для версии 3.11 включительно

Тюнинг кролика – диагностика

```
          Thread      aux check_io emulator       gc    other     port    sleep

Stats per thread:
     async( 0)    0.00%    0.00%    0.00%    0.00%    0.00%    0.00%  100.00%
       aux( 1)    0.03%    0.01%    0.00%    0.00%    0.00%    0.00%   99.96%
dirty_cpu_( 1)    0.00%    0.00%    0.00%    0.24%    0.00%    0.00%   99.76%
dirty_cpu_( 2)    0.00%    0.00%    0.00%    0.00%    0.00%    0.00%  100.00%
dirty_cpu_( 3)    0.00%    0.00%    0.00%    0.00%    0.00%    0.00%  100.00%
dirty_cpu_( 4)    0.00%    0.00%    0.00%    0.00%    0.00%    0.00%  100.00%
dirty_io_s( 1)    0.00%    0.00%    0.00%    0.00%    0.00%    0.00%  100.00%
dirty_io_s( 2)    0.00%    0.00%    0.00%    0.00%    0.00%    0.00%  100.00%
dirty_io_s( 3)    0.00%    0.00%    0.66%    0.00%    0.00%    0.00%   99.34%
dirty_io_s( 4)    0.00%    0.00%    0.00%    0.00%    0.00%    0.00%  100.00%
dirty_io_s( 5)    0.00%    0.00%    0.54%    0.00%    0.00%    0.00%   99.46%
dirty_io_s( 6)    0.00%    0.00%    0.00%    0.00%    0.00%    0.00%  100.00%
```

`rabbitmq-diagnostics runtime_thread_stats`

```
      poll( 0)    0.00%    0.62%    0.00%    0.00%    0.00%    0.00%   99.38%
 scheduler( 1)    0.66%    0.47%    6.10%    1.22%    1.08%    2.36%   88.11%
 scheduler( 2)    0.64%    0.43%    5.59%    1.01%    1.04%    2.05%   89.25%
 scheduler( 3)    0.75%    0.49%    7.74%    1.51%    1.10%    2.57%   85.84%
 scheduler( 4)    0.67%    0.44%    6.13%    0.94%    1.18%    2.41%   88.23%

Stats per type:
         async    0.00%    0.00%    0.00%    0.00%    0.00%    0.00%  100.00%
           aux    0.03%    0.01%    0.00%    0.00%    0.00%    0.00%   99.96%
dirty_cpu_sche    0.00%    0.00%    0.00%    0.06%    0.00%    0.00%   99.94%
dirty_io_sched    0.00%    0.00%    0.22%    0.00%    0.00%    0.00%   99.78%
          poll    0.00%    0.62%    0.00%    0.00%    0.00%    0.00%   99.38%
     scheduler    0.68%    0.46%    6.39%    1.17%    1.10%    2.35%   87.86%
```

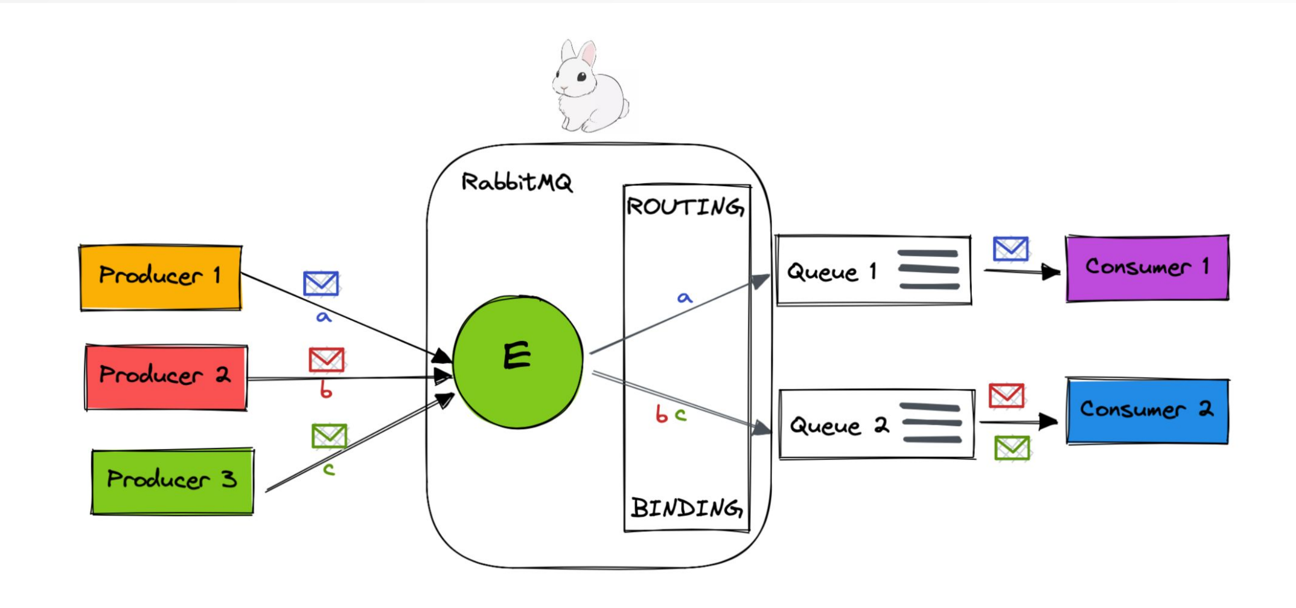


Нужно выбирать с умом

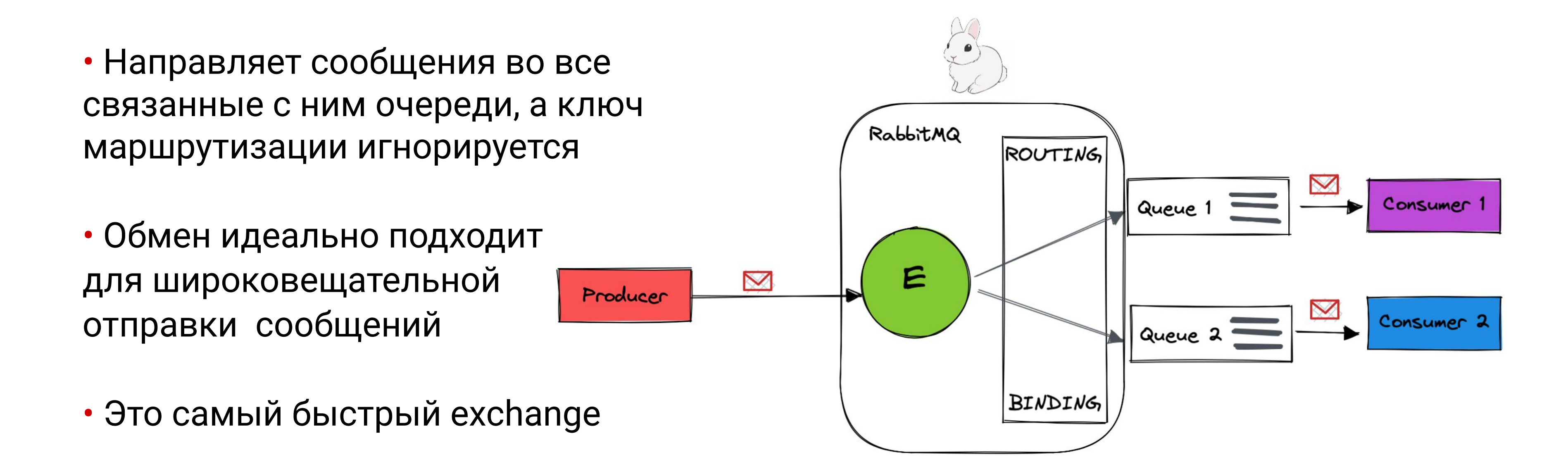
Отклеить на редактуре

Direct Exchange – прямой обмен от сервиса к сервису

- Используется, для одно-адресной маршрутизации
- Поддерживает простой ROUTING

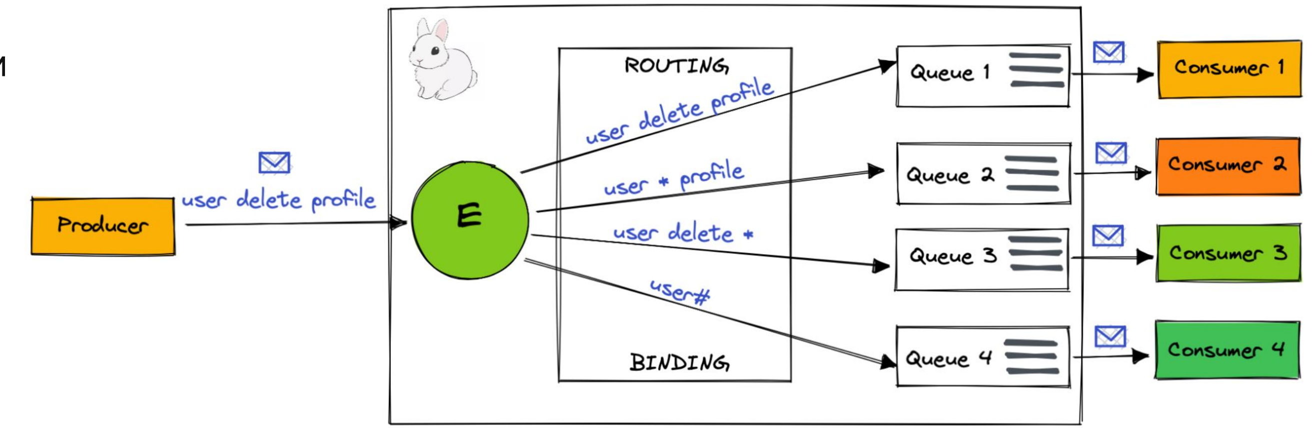


Fanout Exchange – обмен от одного ко многим



Topic Exchange – обмен в рамках общего признака

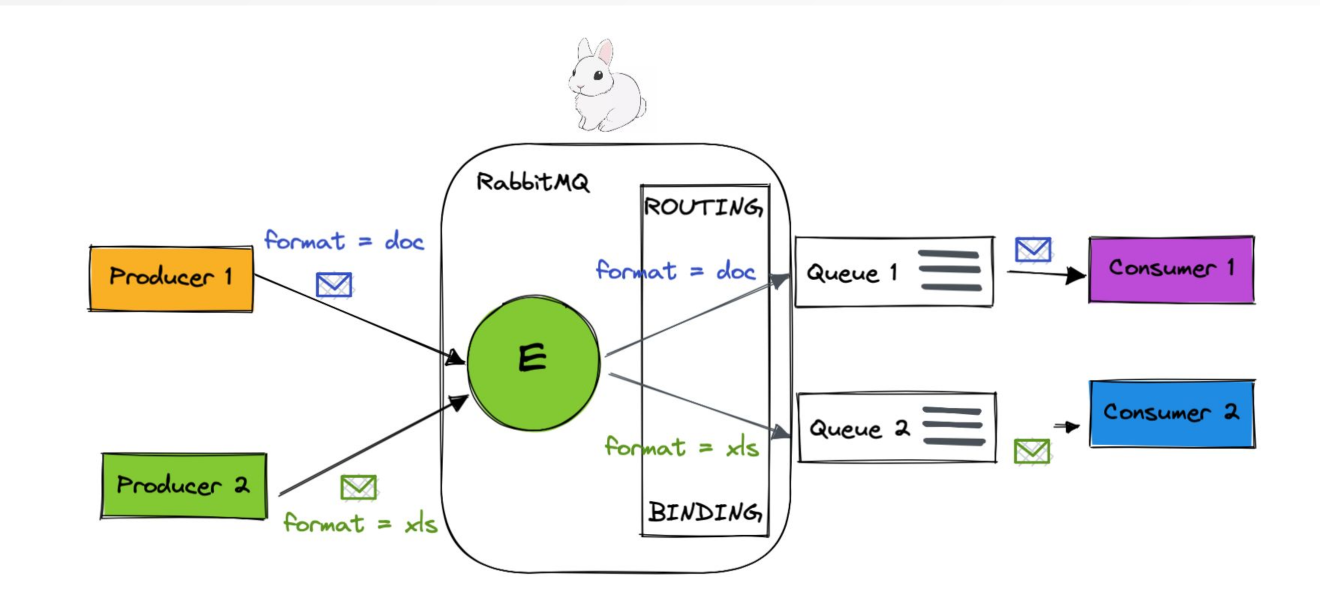
- Используется для обмена сообщениями на основе соответствия ключей маршрутизации
- В роутинге можно использовать спец. символы для более гибкой маршрутизации



Headers Exchange – обмен с использованием заголовков

- Позволяет делать маршрутизацию по нескольким атрибутам которые задаются в заголовках

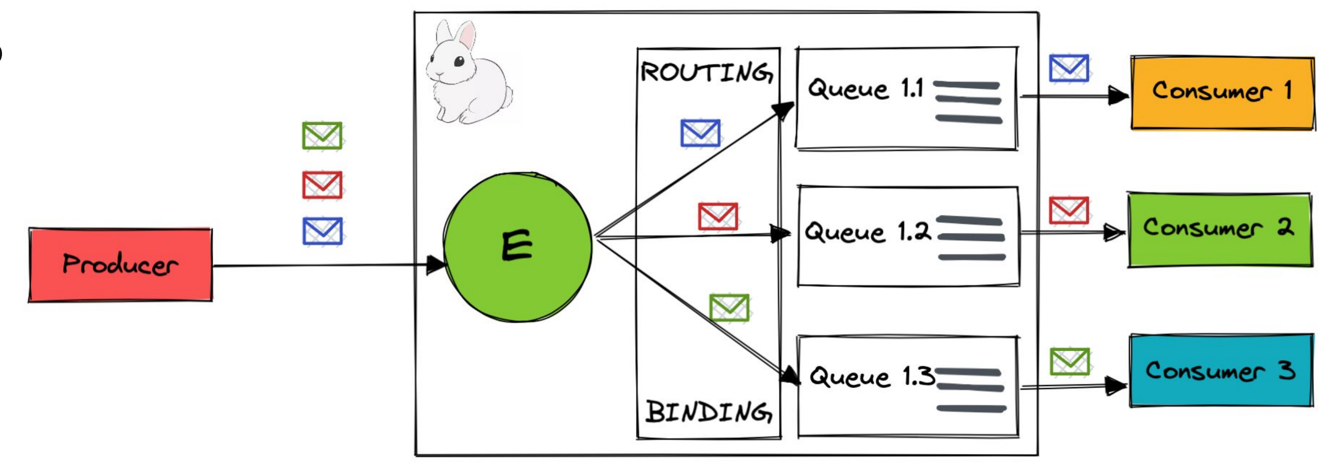
- Игнорирует ключи маршрутизации
  a) дополнительная гибкость
  b) медленнее, чем прочие типы exchange



Consistent-Hashing Exchange (является плагином)

- Предназначен для равномерномерного распределения событий по очередям

- Позволяет увеличить скорость обработки за счет увеличения количества модулей и потоков



Немного о расширениях и плагинах



Немного о расширениях и плагинах



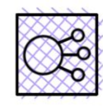
RABBITMQ_AUTH_BACKEND_HTTP



RABBITMQ_MANAGEMENT



RABBITMQ_SHOVEL



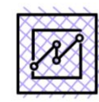
RABBITMQ_AUTH_BACKEND_LDAP



RABBITMQ_PEER_DISCOVERY_CONSUL



RABBITMQ_SHOVEL_MANAGEMENT



RABBITMQ_CONSISTENT_HASH_EXCHANGE



RABBITMQ_PROMETHEUS

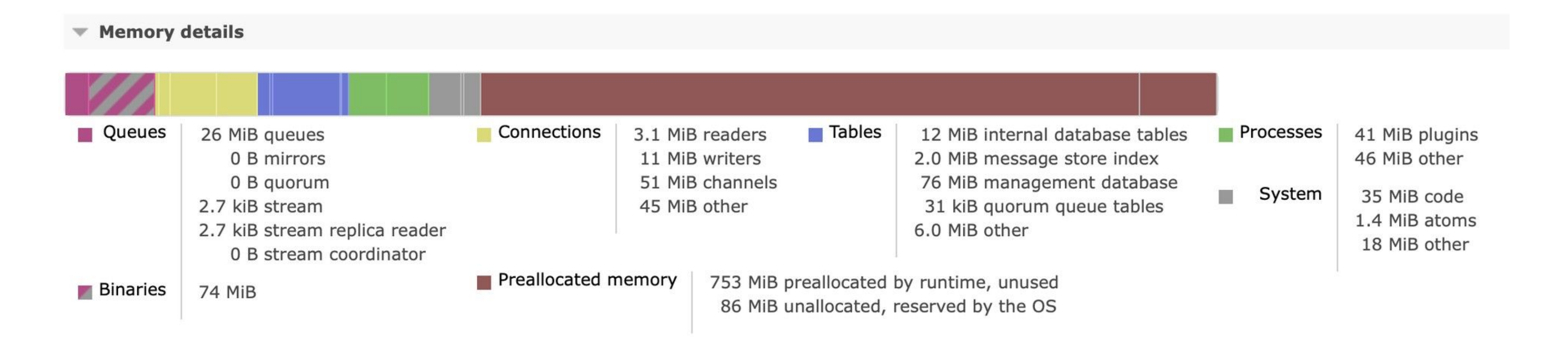
Иконки спёр из AWS!



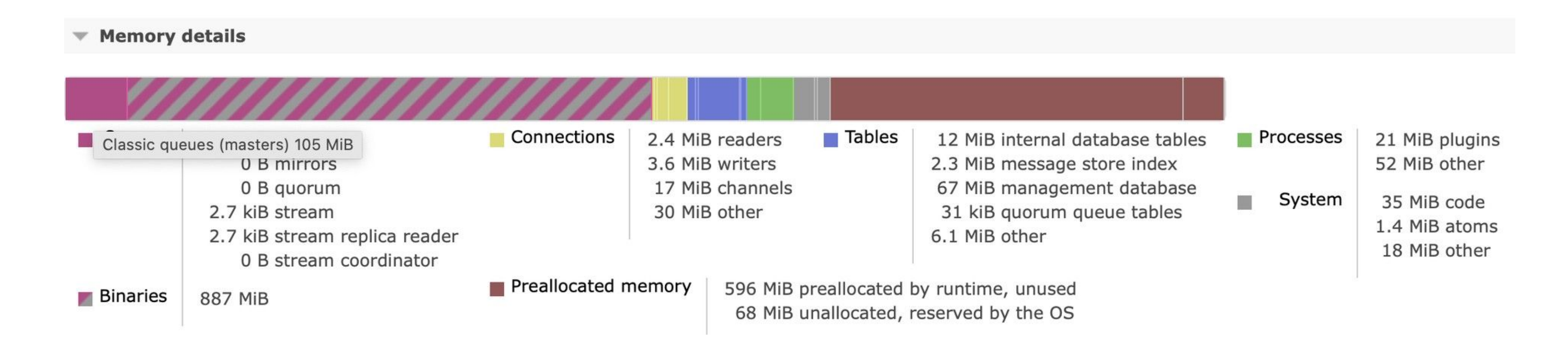


Сгоревший
кролик

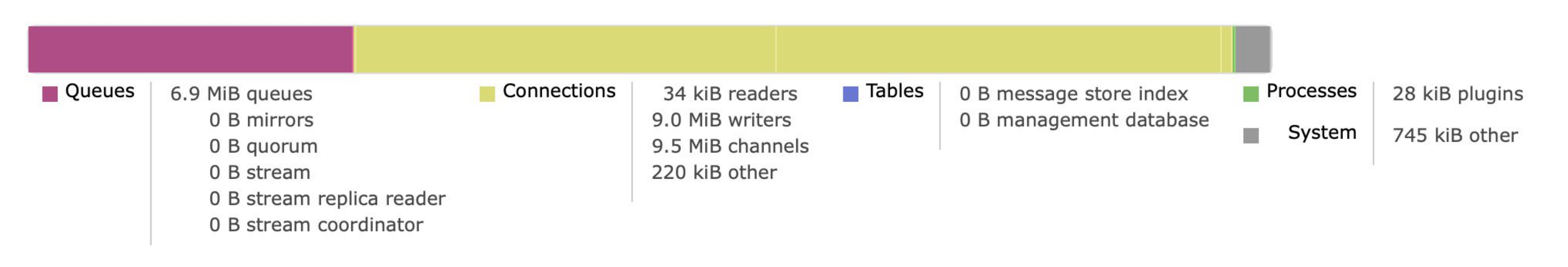
Память в RabbitMq и ее блоки



Блок бинарной памяти и наш провал



Блок бинарной памяти поглубже




```
tcp_listen_options.keepalive = true
```

Watermarks



`vm_memory_high_watermark_paging_ratio = 0.50`

1 — кролик жрет всю память, доступную системе,
может возникнуть конфликт с ОС
0.5 — наш параметр, мы даем 50% доступной памяти.
Этого более чем хватает для бесперебойной работы

`rabbitmqctl set_vm_memory_high_watermark 0.7`

Команда позволяет на лету менять линию ватермарка для памяти

`disk_free_limit.absolute = 5GB`

Не позволяем Кролику съесть все место на диске (оставит 5 гб)

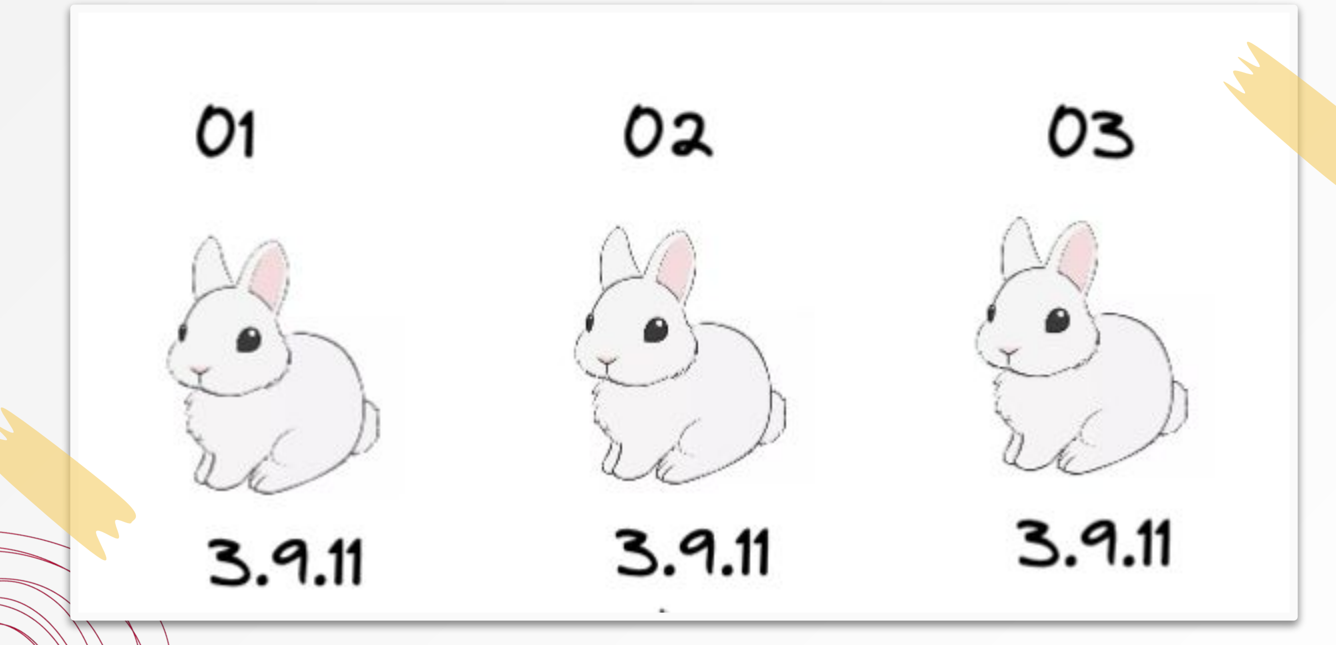
CRUELTY-FREE

Как мы обновляли кластер без простоя

Быстро поднятое упавшим не считается



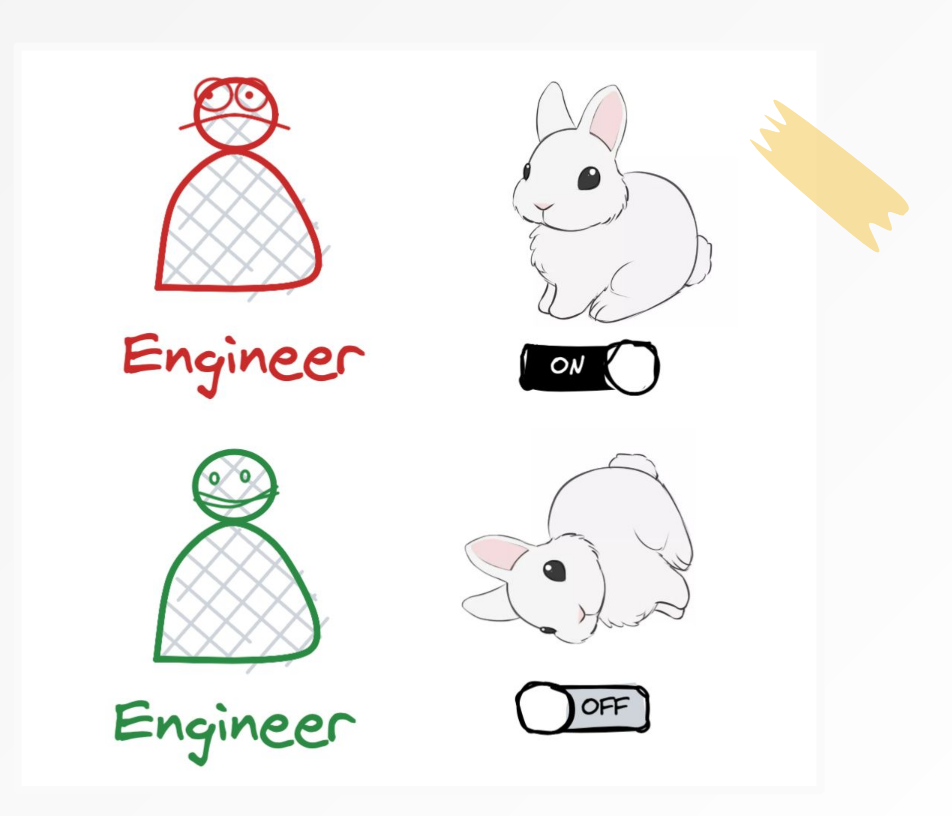
Как мы обновляли кластер без простоя



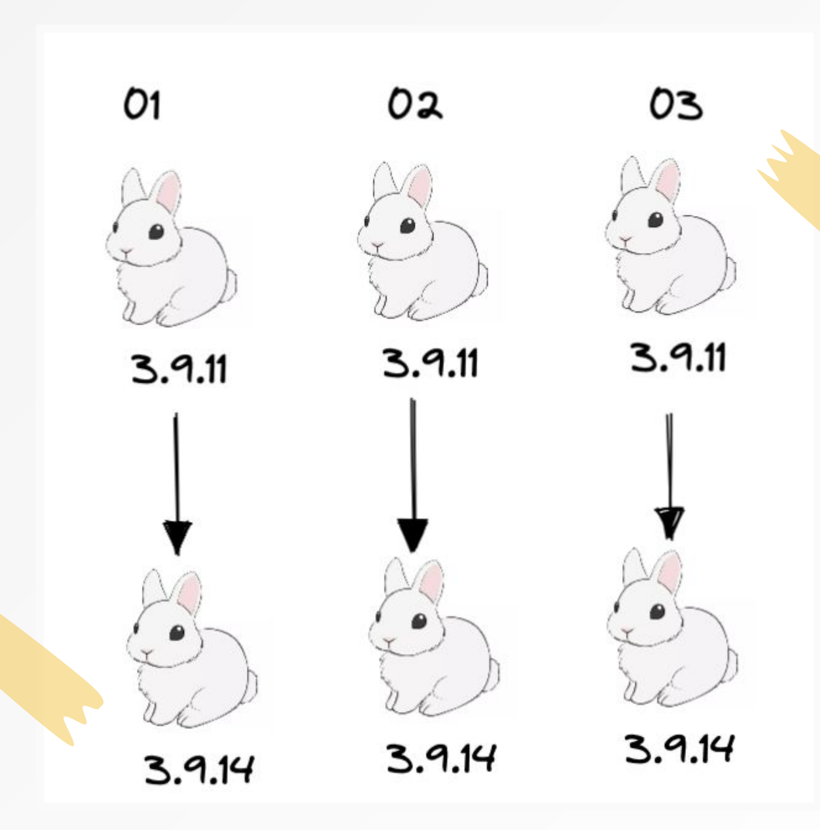
Как мы обновляли кластер без простоя

~ 1 неделя

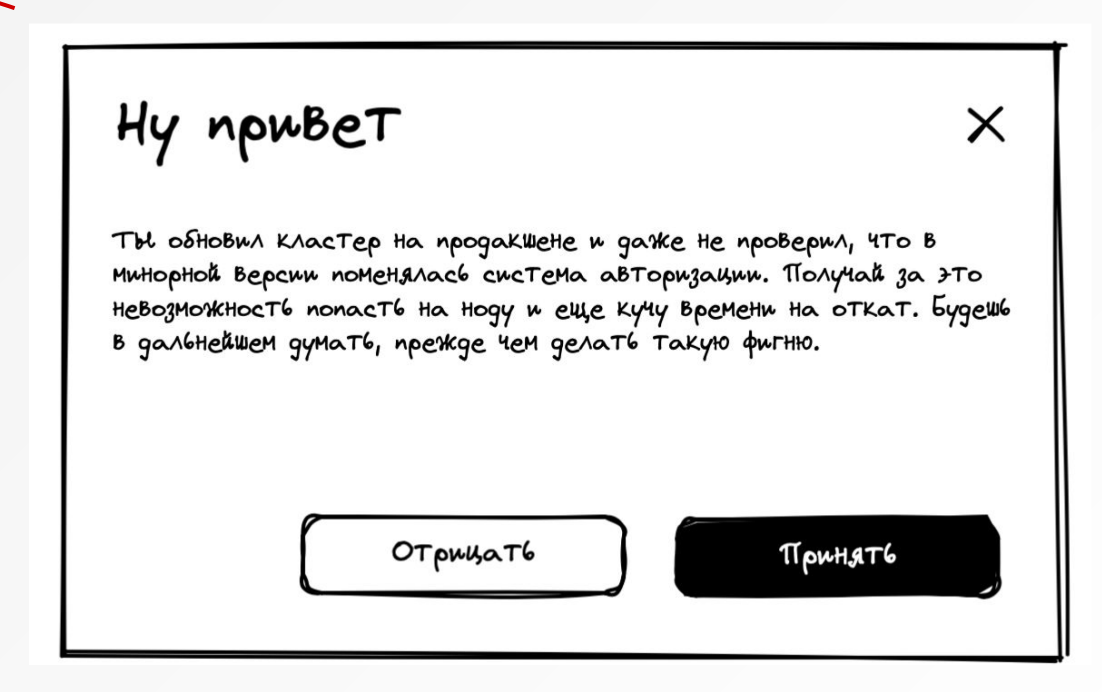
~ 15 минут



Как мы обновляли кластер без простоя



Как мы обновляли кластер ~~без простоя~~



Правила обновлений

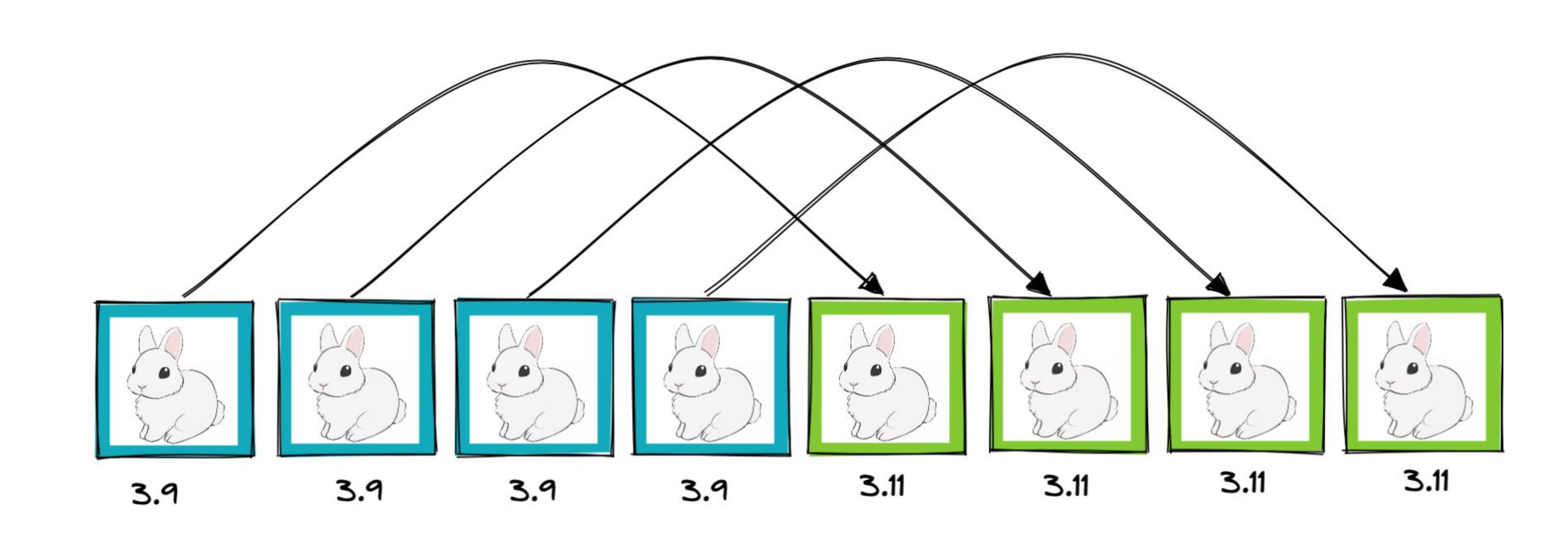


Внимательно читай Changelog!

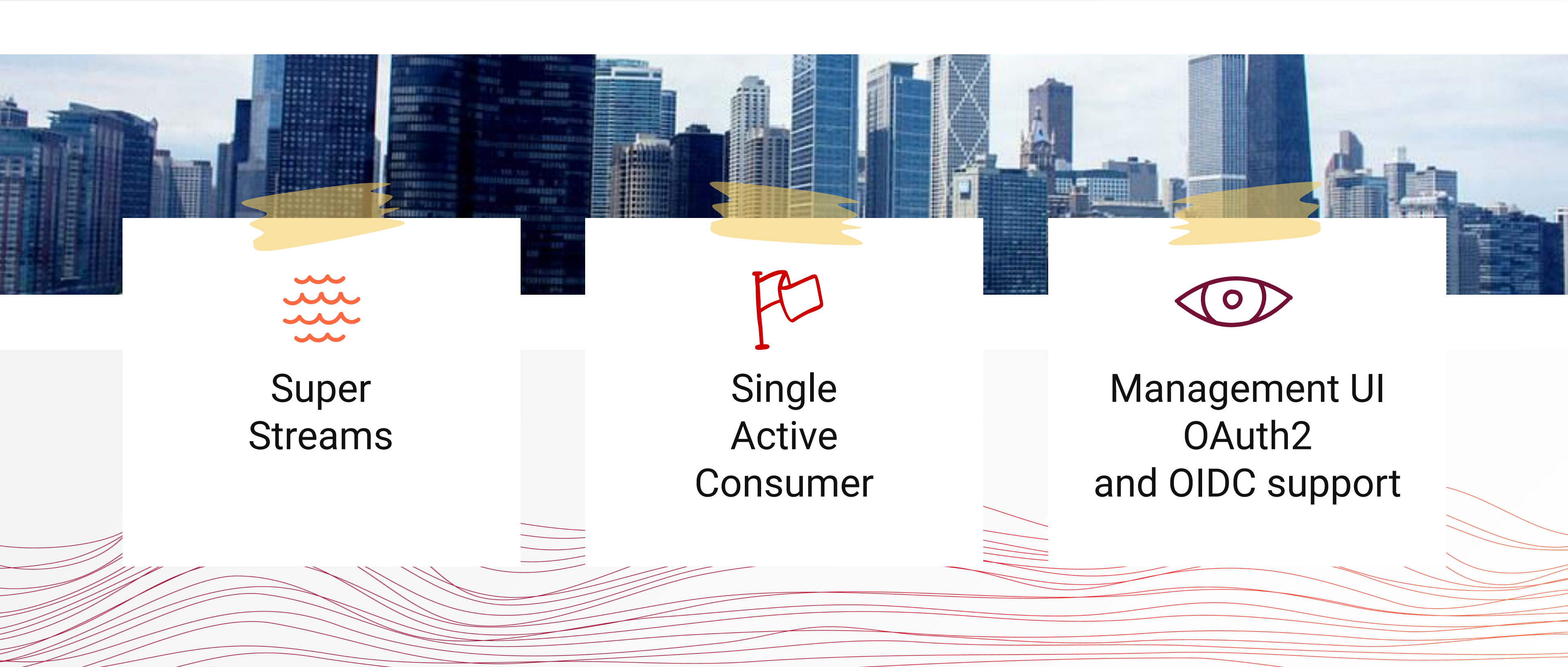
Сначала тестируй на стейдже!

Контролируй зависимости!

Третий вариант — blue-green



Что нового в RabbitMQ 3.11?



Заключение

- Не гонитесь за трендами
- Экспериментируйте
- Вовремя и безопасно обновляйтесь
- Можно добиться высокой производительности и отказоустойчивости на старом добром Кролике
- Ухаживайте за своими кроликами!



Спасибо!

Если есть вопросы, пишите в Telegram!

@Yuruy_Vlasov
@tim256





